A Framework for Scalable Global IP-Anycast (GIA)

Dina Katabi, John Wroclawski

MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139
{dina,jtw}@lcs.mit.edu

ABSTRACT

This paper proposes GIA, a scalable architecture for global IP-anycast. Existing designs for providing IP-anycast must either globally distribute routes to individual anycast groups, or confine each anycast group to a pre-configured topological region. The first approach does not scale because of excessive growth in the routing tables, whereas the second one severely limits the utility of the service. Our design scales by dividing inter-domain anycast routing into two components. The first component builds inexpensive default anycast routes that consume no bandwidth or storage space. The second component, controlled by the edge domains, generates enhanced anycast routes that are customized according to the beneficiary domain's interests. We evaluate the performance of our design using simulation, and prove its practicality by implementing it in the Multi-threaded Routing Toolkit.

Keywords: Anycast, Routing, Scalable, Internet, Architecture

1. INTRODUCTION

IP-anycast is a network service that allows a sender to access the nearest of a group of receivers that share the same anycast address, where 'nearest' is defined according to the routing system's measure of distance. Usually the receivers in the anycast group are replicas, able to support the same service (e.g., mirrored web servers). Thus, accessing the nearest receiver enhances the performance perceived by the sender, saves the network's bandwidth, and provides the desired service. Figure 1 illustrates IP-anycast.

Anycast has numerous potential applications. RFC 1546 [25] proposes anycast as a means to discover a service location and provide host auto-configuration. For example, by assigning the same anycast address to a set of replicated FTP servers, a user downloading a file need not choose the best server manually from the list of mirrors. The user can use the anycast address to directly download the file from the nearest replica.[1] The application of anycast to host auto-configuration, on the other hand, is exemplified in the assignment of the same anycast address to all Domain Name Servers (DNS). In this case, a host that is moved to a new network need not be reconfigured with the local DNS address. The host can use the global anycast address to access the local DNS server anywhere. Recently, IP-anycast has been proposed as an infrastructure for multicast routing. For example, Kim et al. use anycast to allow Protocol Independent Multicast Sparse Mode (PIM-SM) to support multiple rendezvous routers per multicast tree [17], while Katabi uses anycast in designing an intra-domain multicast routing protocol that reduces bandwidth consumption and alleviates traffic concentration [14].

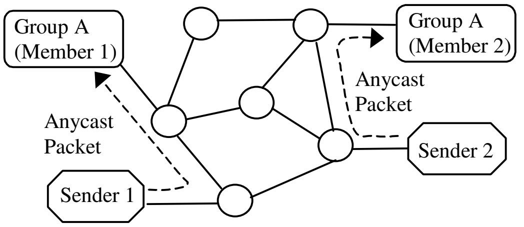


Figure1: Illustration of IP-anycast

Although Sender 1 and Sender 2 are sending to the same anycast address (Group A), the network delivers each packet to the group member nearest its sender.

Currently, there is no scalable design for global IP-anycast. The traditional approach routes anycast addresses using the unicast routing protocols, a design decision that makes anycast unscalable. Unicast routing scales by aggregating routes to destinations that share the same prefix into one routing entry (CIDR [8]). Anycast, on the other hand, defies this form of hierarchical aggregation. An anycast address, like a multicast address, represents a group of nodes that share a particular characteristic and exist somewhere in the Internet. There is no reason to expect anycast group topology to be hierarchical or to comply with the unicast topology. Therefore, routing anycast using the unicast routing protocols requires advertising each global anycast address separately. This requirement causes the routing tables to grow proportionally to the number of all global anycast groups in the entire Internet, and hence does not scale. Figure 2 illustrates anycast's defiance of hierarchical aggregation.

This paper proposes GIA, a scalable architecture for global IP-anycast. GIA scales by capturing the special characteristics of the anycast service in its inter-domain routing protocol, which

---

This research was supported by the US Defense Advanced Research Projects Agency (DARPA) under contract number N66001-98-1-8903.

[1] Methods for the use of anycast for TCP-based services, as in this example, are discussed in [1] and [25].

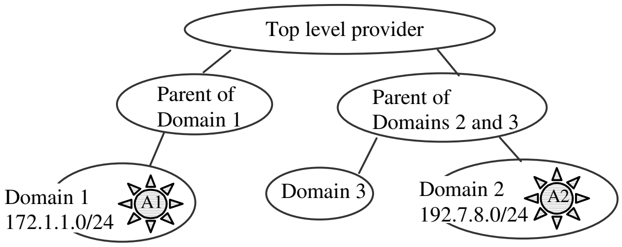


Figure 2: Anycast's defiance of hierarchical aggregation. A1 and A2 are members of the same anycast group. If the group's address shares the prefix with Domain 1 then Domain 2 cannot aggregate the anycast address in its prefix and should advertise it as a separate entry in BGP. A similar situation arises if the group's address shares the prefix with Domain 2.

generates two types of routes: 1) default inexpensive routes that consume no bandwidth or storage space; 2) enhanced shortest path routes that are customized according to the beneficiary domain's interests. Although the architecture is described assuming that a path's length is measured using the unicast measure of distance (number of hops), we show in Section 5 that GIA can use other measures of distance such as average latency or available bandwidth.

The paper is organized as follows. The next section provides a discussion of related work. Section 3 provides a general overview of the design. The details of our address architecture and routing protocols are described in Section 4. Section 5 shows GIA's ability to use a variety of distance measures. Section 6 discusses performance and overhead. A brief description of our implementation of a GIA-enabled border router is provided in Section 7. Deployment is addressed in Section 8. Finally, Section 9 presents our conclusion.

2. BACKGROUND AND RELATED WORK

Anycast was defined in 1993 by RFC 1546 [25]. The document proposes anycast as a means for service discovery and host auto-configuration. It recommends assigning anycast its own address space. It also points out the major difficulties challenging the deployment of IP-anycast. The first difficulty is anycast's defiance of hierarchical aggregation, which makes the service hard to scale. The second difficulty is the stateless nature of the service, an issue that makes the establishment of TCP connections on top of anycast addresses problematic. The RFC proposes a mechanism for establishing TCP connections to anycast destinations; however, it leaves the scalability issue unresolved.

Anycast has been adopted by all proposed successors of IPv4: Pip [6], SIPP [11], and IPv6 [13]. In particular, IPv6 allocates anycast addresses from the unicast address space, making them indistinguishable from their unicast counterparts. Each anycast group is confined to a particular topological region with which it shares the address prefix. Within the region identified by the shared prefix, each member of the anycast group is advertised as a separate entry in the unicast routing system. Outside that region, the anycast address may be aggregated into the routing advertisement for the shared prefix. By confining each anycast group to a predetermined region, IPv6 lessens anycast's scalability problem but does not solve it. Global anycast groups still must be advertised as separate routing entries throughout the entire Internet. These global groups are necessary for many anycast applications, such as the ones in [1,25]. Moreover, they are desirable even in situations where the group members are currently located in a confined region. An example of such situation would be a company that provides an online service and uses an anycast address to have its customers access the nearest online office. Although the company online offices might cover only the US, the company would still want to use a global address, since a scoped anycast address prevents future expansion to Europe and Asia.

Also related to our work are proposals for providing an anycast service at the application layer [2,4,7,22,27]. This approach attempts to build a directory system which, queried with a service name and a client address, returns the unicast address of the server that is nearest the client and that supports the service. Application layer anycast has both advantages and disadvantages over IP anycast. The first disadvantage is that application layer anycast exhibits several complications and scalability problems. More specifically, providing an anycast service at the application layer requires collecting two types of information: 1) information about the servers that are up and supporting a particular service, 2) information about the distance between each potential client and the different servers measured using the metric of interest. To obtain the first type of information, the anycast directory needs either to repeatedly probe the servers or to have the servers repeatedly report their availability to the directory. Given the potentially huge number of services and servers, both mechanisms create a substantial overhead on the network and the directory. Obtaining the distance information is also problematic. For example, if distance is measured by the average network latency, then we need to probe the client from the server or a system collocated with the server to discover the path latency. Similarly, if distance is measured by the number of hops then we need to traceroute the client from the server. In comparison, in IP-anycast, a server's availability is discovered by its local router and the distance information gets updated naturally as part of the routing protocol. Another disadvantage of application layer anycast is its inability to satisfy some classes of anycast applications such as using anycast as an infrastructure for multicast routing [14,17]. A third disadvantage of application layer anycast is its lack of a bootstrap mechanism whereby users access the nearest anycast directory. On the other hand, application layer anycast has two main advantages. First, it is easier to deploy than IP-anycast because it does not involve modifying the routers. Second, it can use distance metrics that are available only at the application level such as the server load. The authors believe that both IP-anycast and application layer anycast deserve further research to fully understand their capabilities and determine their future. Providing a scalable architecture for IP-anycast is a step towards that end.

3. DESIGN RATIONALE

We think the traditional belief that IP-anycast should be routed similarly to unicast has hampered the acceptance and deployment of anycast. The anycast routing protocol should rather recognize the characteristics of IP-anycast and benefit from them to scale. Forcing anycast to obey the unicast routing paradigm wastes routing resources. For example, it is inefficient for a router at a US university (e.g., MIT) to spend equal amounts of routing resources on the route to the Yahoo site and the route to London's Public Transportation site. The first route is used every minute by users in the university, whereas the second one is rarely if ever used. Thus, at a particular edge domain, anycast routes are not equally valuable, and a good anycast routing protocol devotes more resources to frequently accessed anycast groups.

Furthermore, an anycast group represents a network service. In computer systems, it is a common practice to scale services by caching. For example, the Web, a network service, scales by caching repeatedly accessed documents in a community at a local proxy. Similarly, it is likely that at any given time there is a predictable set of anycast groups that users in a domain access with high probability, and that this set is much smaller than all anycast groups in the entire Internet. Anycast can scale by caching at each edge domain routes to groups frequently accessed by the domain's users.

GIA's design allows an edge domain to discover, store and maintain efficient routes to anycast groups repeatedly accessed by users in the domain, while supporting an inexpensive fallback mechanism to send packets to unpopular groups. In fact, the fallback mechanism does not consume any bandwidth or storage space because it is based on mapping the anycast topology to the underlying unicast topology. This design scales for the following reasons. First, it prevents wasting routing resources on rarely used routes, which scarcely affect the perceived performance of the service. Second, by pushing most of the work to the edge domains where routers have small routing tables and many free CPU cycles, it provides a good topological alignment between workload and resources. Finally, because each edge domain spends its routing resources on the anycast routes repeatedly accessed by its own users, it places the workload on the domain that derives the benefits. This creates incentive for edge domains to control the number of their anycast routes to stay within the limits of the available routing resources.

4. DESIGN DETAILS

This section describes the details of the architecture.

4.1 Address Architecture

GIA assigns anycast its own address space. Thus, as illustrated in Figure 3, an anycast address starts with a fixed length bit-pattern that identifies anycast addresses from their unicast and multicast counterparts. We call this prefix the 'Anycast Indicator'.

GIA allocates anycast addresses to domains according to their allocated unicast address space. Hence, the second field in an anycast address is the unicast prefix of the Internet domain that owns the anycast address. We call this field the 'Home Domain Prefix'. It has a variable length that depends on the size of the domain's unicast address space. GIA requires that the home domain contain at least one member of the anycast group. Note that the anycast address is still global and can be assigned to machines anywhere in the Internet.

The last field in an anycast address is the group ID. It has a variable length and it identifies a particular group among the anycast groups that share the same home domain.

The address architecture as it is described above allocates to every Internet domain an anycast address space proportional to its unicast address space.[2] A domain might use its allocated anycast addresses to provide global services. Alternatively, the domain might lease some of its anycast addresses to end users providing online services, or even to other domains. For example, assume *x* is a company that provides an online service and that wants its customers to use an anycast address to locate the online office nearest to them. It is likely that company *x* has a main office connected to the Internet somewhere. Thus, it can use one of the anycast addresses associated with its network for its online service. In this case the home domain for the anycast group is company *x's* domain. On the other hand, if company *x* does not have its own domain, it can lease an anycast address from its service provider, in which case the group's home domain is the provider's domain. In either case, if company *x* grows in the future and opens a new online office, the new office can use the same anycast address and be accessible to customers in its neighborhood.

Anycast Indicator (fixed length)	Home Domain's Unicast Prefix (variable length)	Group ID (variable length)

Figure 3: The syntax of an IP-anycast address

Finally, well-known anycast addresses used for host auto-configuration (such as the group of all DNS servers) should have their home domains in one of the backbones or as virtual domains advertised by the backbones.

4.2 Address Assignment

The process according to which a domain assigns an anycast address to an end user is domain-dependent and can be the same as the one used for assigning unicast addresses. For example, the anycast address might be assigned manually by the administrator or by special address assignment servers that lease anycast and unicast addresses to end users. In addition, the Internet Assigned Numbers Authority (IANA) will assign a set of well-known anycast addresses to a variety of host auto-configuration groups.

4.3 Joining an Anycast Group

To join an anycast group, a host asks its first hop router to advertise the group's address on its behalf. This communication can be achieved by adding a new message type to either the Internet Group Management Protocol [5] or the Neighbor Discovery Protocol [23]. The router advertises the address according to the anycast routing protocol adopted by the domain. It uses a keep-alive mechanism to ascertain the availability of the anycast member, and never advertises an address after the member becomes inaccessible. In addition, the router will likely use a security procedure to ensure that the host is allowed to join the anycast group, as allowing uncontrolled joins to anycast groups creates the potential for a denial of service attack.

4.4 Anycast Routing

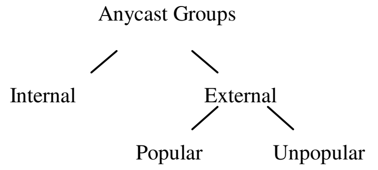
We begin this section by providing some useful definitions, then we describe the details of our routing protocols.

4.4.1 Definitions

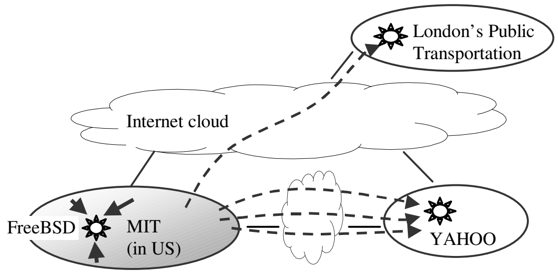
Domain: throughout this paper we use the word 'domain' to refer to a routing domain or an autonomous system (AS).

---

[2] One possible anycast indicator is the bit-pattern '11110'. For the case of IPv4, this choice of the anycast indicator means that domains whose unicast prefix is smaller than x.x.x.x/27 are not allocated any anycast address space. However, domains whose unicast space is smaller than x.x.x.x/27 are a special case and are usually behind a Network Address Translator. These domains can lease or buy anycast addresses from other domains.



(a) Classification of anycast groups at an edge domain



(b) From the perspective of MIT, FreeBSD is internal, Yahoo is popular and London's Public Transportation site is unpopular (the arrows represent anycast packets).

Figure 4: Anycast group classification at an edge domain

Neighborhood of a Domain: the neighborhood of a domain of radius H is the set of domains that are H or fewer domain hops away.

Anycast Group Classification: an edge domain (stub domain) classifies an anycast group according to the following rules, which are illustrated in Figure 4.

- An internal anycast group is a group for which the domain has internally at least one member. Note that all groups are internal to their home domain. However, groups might be internal to domains other than their home domains.
- An external anycast group is a group for which the domain has no local members.
- A popular anycast group is an external group that users in the domain frequently access.

4.4.2 Routing Internal Anycast Groups

GIA routes internal anycast groups the traditional way using unicast routing. Intra-domain routing protocols based on the distance-vector algorithm, such as RIP, intrinsically have the ability to provide an anycast service; if run in a network where the same address is assigned to multiple destinations, they simply route to the nearest one [25]. For protocols based on the link-state algorithm to work correctly, routers should abstain from routing through an anycast address. Figure 5 illustrates an example of this problem. Assuming A is an anycast group, router R1 should not mistake the topology in 5-a for that in 5-b and should not try to route packets sent to R5 through A. To solve the problem, a large cost is assigned to virtual links connecting anycast nodes to their local networks, such that they are not used in building routes unless the anycast node is the destination.

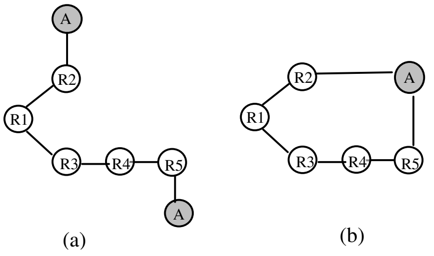


Figure 5: Applying the link state algorithm directly may introduce false topologies

Although routing internal groups using the unicast intra-domain routing protocol causes each internal group to consume an entry in the internal routing table, this approach stays scalable because the number of internal groups is controllable by the domain itself. Therefore, each domain can keep this number within the limits of the locally available bandwidth and storage space.

Finally, in contrast to unicast routing, internal groups are not advertised to other domains in the Internet. Sections 4.4.3 and 4.4.4 describe how users in other domains access those groups. (For those users the groups are external.)

4.4.3 Routing Unpopular Anycast Groups

In GIA, unpopular anycast groups need not be routed. The number of unpopular groups is likely to grow much larger than the number of popular groups. Thus, by using inexpensive default routes to forward packets addressed to unpopular groups, the system, without degrading the service, makes large savings.

A default route does not consume any bandwidth to be generated and does not need any storage space in the routing tables. To understand how such a route exists recall that an anycast address is a concatenation of the anycast indicator, the unicast prefix of the home domain and the group ID. Also, recall that the architecture requires the provider of the anycast service to have at least one member in the home domain. Hence, a router that receives an anycast packet addressed to an unpopular group forwards the packet to the group member in the home domain. To do so the router assigns the destination address to a lookup variable, and shifts the anycast indicator off the variable. After the shifting operation, the address in the variable is a unicast address from the unicast address space of the home domain. The router looks up the variable in its unicast routing table and forwards the packet to the corresponding next-hop, which points towards the home domain. Note that the router leaves the destination address in the packet intact so that other routers that don't have a route for this group may follow exactly the same procedure. Figure 6 shows how an unpopular anycast group is mapped to a unicast address in its home domain.

Thus, a packet addressed to an unpopular group is forwarded towards its home domain. However, depending on the popularity distribution of its corresponding group, the packet follows one of three possible paths. First, if the packet crosses any domain that contains a member of the anycast group then the packet is

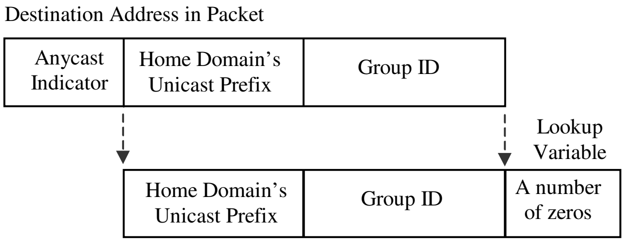


Figure 6: Mapping an unpopular anycast group to a unicast address in the home domain. The anycast destination is assigned to a lookup variable. The anycast indicator is shifted off the variable, then the variable is looked up in the router's unicast routing table. The anycast packet remains intact.

delivered to that member by the intra-domain anycast routing protocol. Second, if the packet crosses a domain that has this group as a popular group and consequently knows a shorter route to one of the group's members, then the packet continues its journey along the shortest path. Finally, if neither of the aforementioned cases is encountered, then the packet eventually hits the home domain and is delivered to the member there.

4.4.4 *Routing Popular Anycast Groups*

At the core of GIA's architecture is generating shortest path routes to popular anycast groups. We begin by giving a general overview of this process, and we provide the details in the next sections. To generate routes to popular groups, the border routers in an edge domain decide which groups are popular in their domain. This decision is made according to the route's level of usage or the domain's policy. Periodically, border routers search their neighborhood looking for the nearest members of popular anycast groups. Once they find the shortest path route to a popular anycast group, they cache the route and tunnel all subsequent packets to the domain where the nearest member resides.

In contrast to unicast inter-domain routing, which is based on advertising unicast prefixes to all Internet domains, GIA adopts an on-demand query-based inter-domain routing protocol. We choose a query-based protocol for two reasons. First, we want a design in which routers in the core of the Internet do not store any anycast routes.[3] Second, the fact that an anycast group is replicated in multiple domains in the Internet increases significantly the probability of finding the nearest group member by exploring a small neighborhood around the interested domain. (Section 6.1 shows that this increase is exponential.)

Our route learning process makes use of the TCP connections a BGP router has with its peers [26]. It involves adding two new messages to BGP: a search message and a reply message. In the following sections we describe the steps of learning anycast routes.

4.4.4.1 *Initiating a search*

To discover which groups are popular in their edge domain, the border routers observe the number of packets recently sent to each anycast group. In addition, the border routers might be configured to consider certain groups as popular regardless of their access level. For example, the BRs in a domain that has no DNS server might be configured to consider the group of all DNS servers as a popular group regardless of its access level. Note that because all anycast packets addressed to a particular group exit the domain at the same border router, each BR decides on the access level of the anycast groups it sees without contacting the other BRs.[4]

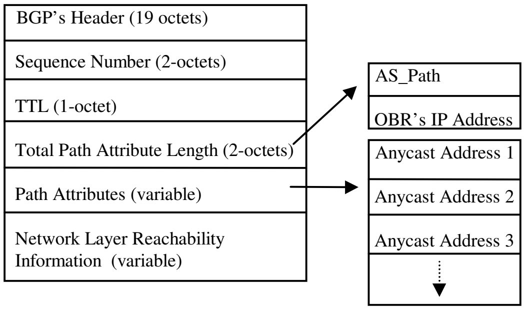


Figure 7: The Format of the search message

A search for a popular anycast group is triggered by the exit border router towards the group's home domain, which receives the anycast packets in the absence of a learned route. We call this border router the originating border router (OBR). At the beginning of each 'Search Interval', the OBR generates a search message for all of the popular groups for which there is no learned route and broadcasts it to all of its peers. The duration of the 'Search Interval' decides the maximum search rate and can be agreed upon with the domain's provider. Once the search is generated the OBR sets a timer and waits for replies. Note that during the search process the OBR does not keep the arriving anycast packets until a route is learned. It forwards them along the default route.

The search is a scoped domain-by-domain broadcast that explores the neighborhood around the searching domain looking for members of popular anycast groups. The search message has the format shown in Figure 7. The message has fields similar to a BGP update. In particular, it contains a path-vector field, which collects information about the autonomous systems the search crosses and prevents the search message from looping. (To comply with BGP's terminology, Figure 7 refers to the path-vector as Path Attributes.) In addition the message contains a TTL field, which scopes the search to a neighborhood around the searching domain. This field is initialized to the maximum number of domain hops the search can traverse, and gets decreased with each domain hop. Note that one search message may solicit routes for many popular anycast groups.

4.4.4.2 *Receiving a search*

The rules for processing a received search message are illustrated in Figure 8. A search message needs to be processed only once in each domain; thus, a border router (BR) that receives a search from an internal peer propagates the message to all of its peers with no further processing.[5] A BR that receives a search message from an external peer examines whether the domain has routes to the anycast addresses in the message. The BR can reply for two types of groups: internal groups, and popular groups for which it has already learned routes. For all groups that are internal to the replying BR's domain, the BR sends a reply message, which relays the path-vector in the original search message after appending the receiving BR's autonomous system number (AS number). In addition, the reply includes the original search sequence number and the receiving BR's IP-address. The reply is sent directly to the OBR.

---

[3] This objective makes it hard to design an advertisement-based routing protocol, because these protocols prevent flooding by storing at each router the shortest route seen so far. Consider unicast routing as an example. If routers in the backbones do not store unicast routes then any insignificant change in the topology will be flooded to all domains in the Internet because the upstream routers cannot tell whether the change in the topology would affect the forwarding path at downstream routers. This flooding effect would be exacerbated by the fact that an anycast address has a virtual high connectivity caused by its replication [18].

[4] In case the border routers are too busy to monitor the access level of anycast groups, a separate device attached to the same link as the border routers can do the job.

[5] Since all BRs in a domain have the same routing table, a search needs to be processed only once in each domain.

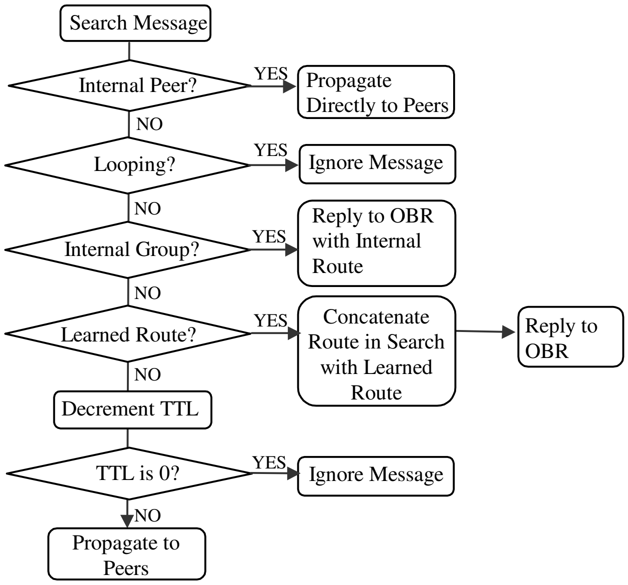


Figure 8: Receiving a search message (simplified by assuming the search message contains 1 address)

Groups for which no internal member is found are looked up in the set of learned anycast routes. For each anycast group for which the receiving BR has a learned route, it concatenates the path-vector in the search with the path-vector of the learned anycast route, and sends the resulting path-vector to the OBR in a reply message. The replier field in the reply message is set to the IP address of the router from which the cached route has been learned. All addresses for which the receiving BR is able to send a reply are removed from the search message. If the message still contains anycast groups for which no route has been found, the receiving BR decrements the TTL of the search, checks that the TTL did not reach zero, and propagates the search to all of its peers.

To prevent a search message from looping, GIA requires a BR that receives a search message whose path-vector includes its own AS to ignore the message. Moreover, to reduce the number of messages spawned by a search, we require each BR to maintain a table of all triples (OBR, sequence number, shortest path-vector for this combination of OBR and sequence number) seen recently (e.g., in the last two average search intervals). A BR propagates a search only if it contains a path-vector shorter than the shortest path-vector with the same (OBR, sequence number) seen so far. Although storing a table of the above-mentioned triples consumes some memory at a border router, the size of the memory needed is relatively small because it is on the order of the number of OBRs in a neighborhood. In addition, the lookups in this table are not on the critical path of unicast data packets.

4.4.4.3 Receiving a reply

After sending a search message, an OBR sets a timer and waits for replies. When the timer expires, the OBR checks all the received replies and chooses the one with the smallest path-vector (the shortest route). The OBR checks its list of pending popular addresses and deletes any address for which it has found a route. The groups the router searched for but for which it couldn't find the nearest members have their popularity multiplied by a decaying factor to reduce their chance of being included in a future search.

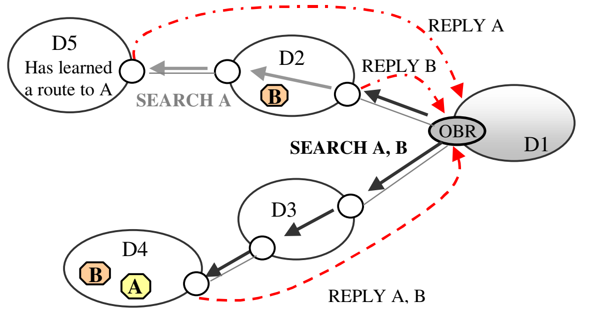
The learned routes are kept in a cache of popular anycast routes. Also, the routes are advertised to all internal peers as if they were learned from a BGP update message. Depending on the domain's policy the routes might be injected into internal routers' routing tables or kept only at border routers.

A stored external anycast route contains the path-vector, which lists the set of domains the route traverses, and the unicast address of the destination BR. Both entries are extracted from the reply message. The path-vector is used in answering search messages issued by neighboring domains looking for a route to this anycast group. The unicast address of the destination border router is used to tunnel all subsequent anycast packets to the domain that has the nearest group member. Figure 9 illustrates an instance of the route learning protocol.

4.4.4.4 Scoping a search

GIA scopes a search such that it is likely to find the nearest group member without flooding the Internet. We do so using two mechanisms. First, a domain that generates a search controls the size of the searched neighborhood by setting the TTL field in the search message. This field should be set such that the search can reach the core of the network. Given that virtually all domains are less than 3 domain hops from the core of the Internet [3], we recommend setting the TTL field to 2 or 3.

Second, transit domains control the scope of a search by instructing their border routers (BRs) not to propagate search messages to distant peers, where the word 'distant' refers either to geographical distance or poor connection. Figure 10 shows an example of a provider network that connects Europe with the US. The BRs in each continent do not propagate searches to the BRs in the other continent. Pruning such searches does not significantly



1. The OBR in domain 1 broadcasts to its peers a search message soliciting routes for popular groups A and B.
2. By inspecting its routing table, the BR in domain 2 discovers that there is a member of B in its domain, so it sends to the OBR a reply for B. However, since it has no route for A, it propagates to its peer the remainder of the search message, which contains only a search for A.
3. The search for A, hits the BR in domain 5, which has a learned route for A. So it concatenates the path-vector in the search message with the learned route it knows, and sends the resulting route to the OBR.
4. The BR in domain 3 does not have a route to either group so it propagates the search to its peer. When the search hits the BR in domain 4, the router recognizes both groups as internal to its domain and sends a reply for both A and B to the OBR.

Figure 9: Learning routes to popular anycast groups

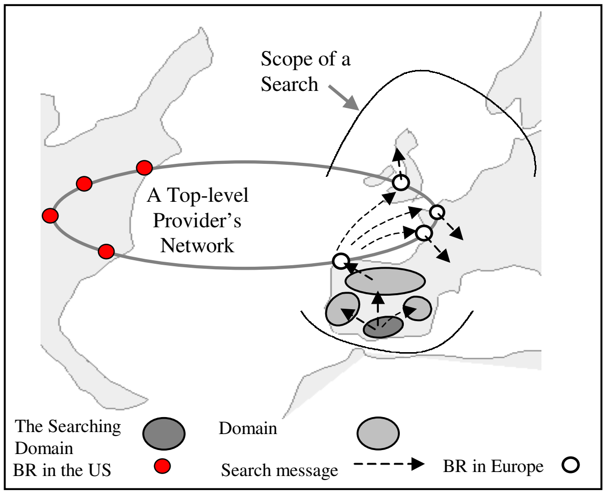


Figure 10: Scoping a search by transit domains

A high level provider that connects Europe with the US instructs the border routers in Europe to propagate searches only to their peers in Europe, and the border routers in the US to propagate search messages only to their peers in the US.

affect performance because the anycast members they would have found are by definition behind a long or congested path. Hence, it is unlikely that these members would have shown a better performance than the member in the home domain. In fact, the role the top-level providers play in limiting the number of messages spawned by a search is essential. Searches spring from edge domains, which have few connections to the rest of the Internet. Thus, in its first hop, a search generates an insignificant number of messages. Only when a search hits a domain with extremely high connectivity does it spawn a large number of messages. These domains represent the networks of the top-level providers. They usually have high connectivity because they span a large geographical region. Thus, by instructing their border routers to propagate searches only to local peers, top-level providers prevent a search from flooding a large part of the Internet. The information necessary to distinguish distant peers is easily available to any provider. It is scalable because it is on the order of the number of border routers in the ISP's network.

4.4.4.5 Withdrawing and Replacing a Learned Route

A learned route becomes invalid in the following cases. First case happens when the domain loses connectivity to the nearest member. In this case, the BGP component of the OBR receives a withdraw message and discovers the loss of connectivity. This causes the OBR to withdraw the learned anycast route and schedule a new search. The second case happens when the nearest anycast member crashes or leaves the group. In this case the domain cannot directly discover the invalidity of the route, and keeps tunneling the packets to the learned BR. However, when those packets arrive at the destination domain, the receiving BR discovers that there is no local anycast member. Thus, it forwards the packets according to its best knowledge of the route,[6] and sends an ICMP message to the BR that tunneled the packet informing it of the invalidity of the learned route. A BR that receives such an ICMP message treats it similarly to a route withdrawal received via BGP. Finally, as a consequence of the route being withdrawn, the OBR schedules the group to be considered in a future search.

On the other hand, a learned route might be withdrawn even when it is still valid so as to allow caching of new popular anycast routes while maintaining an upper bound on the number of cached anycast routes. This can be done by having the exit BR toward the destination domain check the routes' level of usage and withdraw learned routes that are no longer popular.

Finally, to ensure that a learned route remains the shortest route available to its group, a route that stays in the cache longer than a threshold triggers a new search whose TTL is set to one hop less than the current route's length. The current route is preserved unless a better one is found.

5. USING DISTANCE METRICS OTHER THAN HOP COUNT

In the previous sections, we described GIA assuming that distance is measured using the same measure as unicast routing. However, GIA's architecture can use a variety of distance measures such as the average latency, available bandwidth, or number of hops. To do so, an Internet Service Provider occasionally measures the distance between all pairs of border routers in its domain using the desired metric (e.g., average latency, number of router hops, *etc*.). At each border router, the ISP stores the distance from this router to the other border routers in the ISP's network measured using the new metric. Then, a search message collects this information and measures the path length using the desired metric. Note that this approach stays scalable because the measurements are performed locally (to the ISP's network), and the information stored at the border routers is on the order of the number of border routers in the ISP's network.

6. PERFORMANCE

This section uses simulation and discussion to study the performance of GIA. The main results of this section can be summarized as follows. First, although GIA does not provide hard guarantees on accessing the nearest member of an anycast group, on average, the path length in GIA is remarkably close to the path to the nearest member of an anycast group. Second, the growth in the routing tables at each domain is limited and controllable by the domain itself. Third, the processing overhead, mainly located at border routers, allows the current Internet to support millions of global anycast groups. Moreover, the future growth of the Internet will not degrade the service nor will it hinder its scalability.

6.1 Simulation Environment

We implemented a custom simulator to study the performance of GIA. For our simulation topology, we use a set of snapshots of the Internet inter-domain topology generated by NLANR based on the BGP routing tables [24]. Complete information about the simulation topology is provided in Appendix A1.

In the absence of an anycast service from the current Internet, there is no data about the usage or characteristics of anycast groups. Therefore, we had to use some assumptions to carry out our simulations. We believe our assumptions are conservative and

---

[6] Most likely the BR will forward the packets to their home domain. However, it might be the case that after the local anycast member crashed, the domain has learned a route to some other nearby member.

that simulations using them tend to provide a lower bound on the efficiency and an upper bound on the overhead.

First, for each anycast group we choose the home domain randomly from all the domains in the Internet. We randomly assign members of an anycast group to domains. However, if the percentage of domains that have members of an anycast group is less than 1% then we do not assign two members of that group to adjacent domains. We consider any domain with one or two connections to the rest of the Internet as an edge domain. This means that around 75% of the domains in the Internet are edge domains. The above-described policy for assigning members of an anycast group is conservative because it ignores the fact that providers of anycast services tend to establish servers in network regions where their services are popular. Moreover, it assigns group members with equal probability to isolated domains and to well-connected domains.

Second, we model the popularity distribution of anycast groups after the popularity distribution of web servers. We choose this model because, among the currently proposed anycast applications, locating mirrored web servers is the most resource consuming. Thus, it is likely to be the application stressing the scalability of the anycast service. The data for the popularity distribution of web servers is from the organizational trace in [28]. It studies a weekly trace from 175 different organizations accessing 995374 web servers. To use this data, we scale the number of organizations to the number of domains in the Internet, and the number of web severs to the total number of global anycast groups in our simulation. Note that because the web trace shows only servers that are accessed by one or more of the organizations, the model is biased towards increasing the number of popular groups and consequently increasing the search overhead in GIA.

The average lifetime of a learned anycast route depends mainly on the average period a group stays popular at an edge domain. This parameter can be modeled after the lifetime of a document at a web proxy, which is around 50 days [10]. Other parameters such as changes in the external unicast routes used in mapping the cached anycast routes decrease the lifetime of a popular anycast route. Hence, we assume that the average lifetime of a learned anycast route is 30 days. Nonetheless, we point out that the effect of external unicast route changes is negligible because edge domains have few links to the rest of the Internet and significantly stable external unicast routes.

According to Section 4.4.4.4, border routers in an ISP's network do not propagate search messages to distant peers, where the word 'distant' refers to geographical distance or poor connection. Although this information is locally available to an ISP, it is not supported by the Internet graphs we were able to obtain. Therefore, we simulate the search scoping by transit domains using simpler rules. We compute the number of shared neighbors between any pair of the 5% most connected domains in the Internet. A highly connected domain receiving a search message from another highly connected domain does not propagate the search if more than 1% of its neighbors are shared neighbors with the upstream domain. Also, a domain propagates a search between two highly connected domains only if the percentage of neighbors they have in common is less than 3% of the downstream domain's neighbors. The intuition here is that most of the top-level providers have similar connectivity (e.g., they connect the US East Coast to its West Cost). Therefore, when a top-level provider propagates a search received from another top-level provider, it generates many redundant messages, which the above rules prune. We believe these rules are fairly simple and can be exercised easily by any ISP. The information necessary to support these rules is publicly available at [24]. Also, it can be gleaned locally at any ISP from the BGP routing tables. Note that this model tends to overestimate the number of search messages because transit domains do not exercise full control over the scope of a search.

Finally, our simulator does not have the ability to simulate learning routes from domains that have cached routes. Hence, the simulator tends to underestimate the performance and exaggerate the overhead of the protocol. Routes to highly popular anycast groups would generate considerably fewer messages in practice than in our simulation. This happens because each simulated search ignores the fact that there are many domains around that can stop the search and reply with a cached (learned) route. In fact, the large size of the Internet topology has rendered simulating learning a cached route a computationally exhausting task. However, the inability of our simulator to benefit from learning cached routes could be regarded as an additional factor in making the simulation environment conservative.

6.2 Efficiency of the Path Computed by GIA

We measure the efficiency of the path computed by GIA by comparing it against the shortest path, where the term 'shortest path' refers to the path computed by routing anycast the traditional way via unicast routing.

Since internal anycast groups are routed using unicast routing, the path computed by GIA to internal groups is the shortest path. On the other hand, the path to external anycast groups, on average, is longer than the shortest path. The difference is due to the existence of packets addressed to unpopular groups and to the possibility of a search failure.[7] Assuming that $R_{home/nearest}$ is the average ratio of the path length to the home member to the path length to the nearest member, $R_{popular/nearest}$ is the average ratio of the length of the path used by GIA to access a popular group to the length of the path to the nearest member of that group, and ϱ_{pop} is the percentage of anycast traffic at an edge domain that goes to its popular groups, then the average ratio of the external path used in GIA to the shortest path can be computed as follows.

$$R = Average\ (external\ path\ in\ GIA\ /\ shortest\ path)$$

$$R = \varrho_{pop}\ R_{popular/nearest} + (1 - \varrho_{pop})\ R_{home/nearest}$$

Taking into consideration that when a search for a popular group fails GIA ends up using the home member, and assuming $p_{success}$ is the probability a search succeeds in finding the nearest group member, and $R^{*}_{home/nearest}$ is the average ratio of the path length to the home member to the path length to the nearest member given that the search has failed, the value of $R_{popular/nearest}$ can be written as follows.

$$R_{popular/nearest} = p_{success} \times 1 + (1 - p_{success}) R^{*}_{home/nearest}$$

By substitution,

$$R = \varrho_{pop}\ (p_{success} + (1 - p_{success})\ R^{*}_{home/nearest}) + (1 - \varrho_{pop}) R_{home/nearest}$$

---

[7] The search fails when the nearest member of the popular group is outside the searched neighborhood.

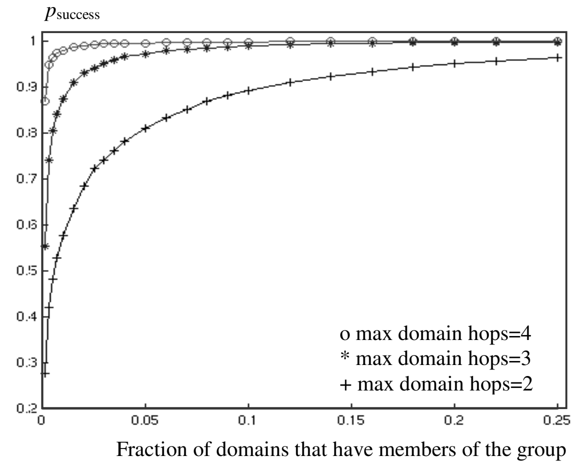


Figure 11: The probability a search finds the nearest member of a popular group as a function of the fraction of domains that have members and the maximum number of domain hops in the TTL field

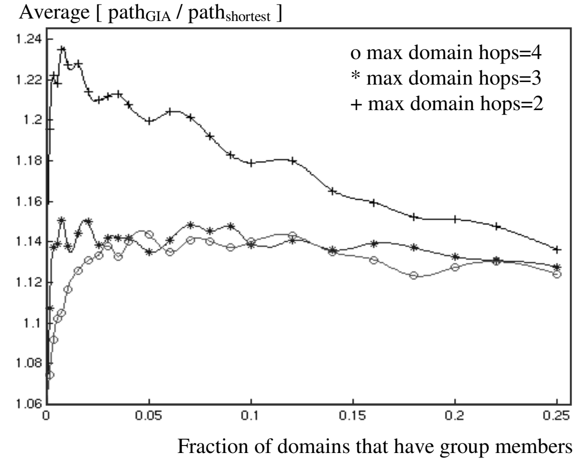


Figure 12: The average of the ratio of the path length in GIA to the shortest path as a function of the fraction of domains that have members and the maximum domain hops in the TTL field

We estimate the parameters in the above equation by simulating GIA using the graph of the Internet inter-domain topology of November 1999 [24]. The simulation environment is described in the previous section. Each data point in our graphs is the average of 100 runs.

Figure 11 estimates $p_{success}$, the probability a search succeeds in finding the nearest anycast member, as a function of both the maximum number of domain hops for which we propagate a search, and the fraction of domains that have members of the anycast group. Note how $p_{success}$ increases exponentially with the increase in the fraction of domains that have members of an anycast group.[8] This high probability of a search success implies a reasonable load balance among the members in an anycast group because in each neighborhood clients are locating and accessing their local server.

Figure 12 shows the ratio of the external path length in GIA to the shortest path (R) as a function of both the maximum number of domain hops for which we propagate the search, and the fraction of domains that have members of the anycast group. The values of $p_{success}$, $R_{home/nearest}$ and $R^*_{home/nearest}$ are found from simulation, whereas the value of ρ_{pop} is assumed to be 80%.[9] Note that as the fraction of domains that have members of the anycast group decreases ($x \rightarrow 0$), GIA's path approaches the shortest path. This is expected since when there is only one member in the anycast group, it has to be in the home domain. Similarly, when the fraction of domains that have members increases ($x \rightarrow 1$), GIA's path approaches the shortest path because all searches succeed. Moreover, all anycast packets forwarded along a default route to an unpopular group immediately hit an adjacent domain that has an internal member of the group and get delivered to that member.

The simulation indicates that it is sufficient to search a neighborhood of 2 to 3 domain hops to observe a good performance. In particular, if the search is sent to a maximum of 3 domain hops then the average path in GIA stays within 1.15 of the shortest path. This high efficiency is a natural result of the fact that the probability of a search success increases exponentially with the number of domains that have members. It also results from the fact that the diameter of the Internet is only 10 domain hops. (It has been 10 domain hops for the past 6 years and is unlikely to change [3,9].) Therefore, even for the case of unpopular groups when we send the packets to the home domain, on average the home domain is only 5 domain hops away.

Although the 2-domain-hops curve in Figure 12 shows a worst case inefficiency of 1.23, the occurrence of the worst case behavior is unlikely in practice. The worst case behavior happens in our simulation when the fraction of domains that have group members is less than 1%. Yet, recall that our simulation assigns group members randomly to domains. Given that most of the domains are edge domains, members are likely to end up in an isolated part of the graph. For groups with considerably few members, this isolation causes a remarkable decrease in the efficiency. However, in practice, groups with few members and widely spread customers are likely to locate their members in highly connected domains. Thus, we still think that sending the search to a maximum of 2 domain hops results in an acceptable efficiency. The choice whether to use a search radius of 2 or 3 should be made by the edge domain depending on how far it is from the core of the Internet.

6.3 GIA's Effect on the Routing Tables

In contrast to the traditional approach for IP-anycast, where the routing tables grow proportionally to the number of all global anycast groups, the growth in the routing tables in GIA is manageable. In particular, routers in the backbones, which usually

---

[8] The exponential increase in $p_{success}$ can be understood by the following argument. Assume that the percentage of domains in a searched neighborhood is x, then for a group of y members (spread in different domains) the probability none of them is in the searched neighborhood is $(1-x)^y$. The probability the search succeeds is equal to the probability that at least one of the group's y members is in the searched neighborhood. Thus, it is given by $1-(1-x)^y$.

[9] We think 80% is a reasonable value for ϱ_{pop}. However, using ϱ_{pop} = 70% results in similar graphs to those in Figure 12, yet shifted up by 0.05.

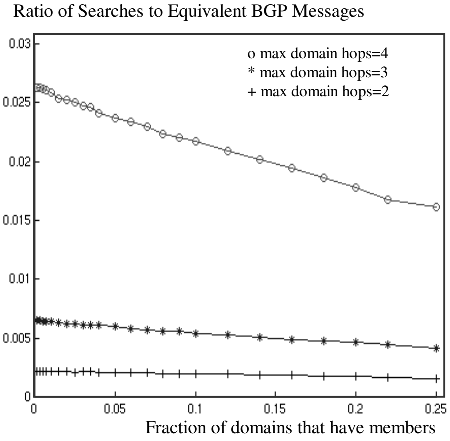


Figure 13: The ratio of the number of search messages generated in GIA to the number of messages generated by routing anycast using non-GIA- enabled BGP.

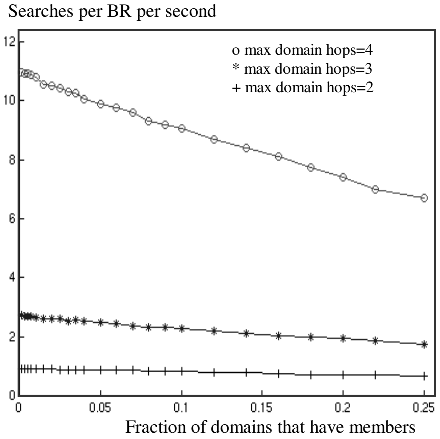


Figure 14: The Average number of search messages processed by a boarder router per second assuming that the number of global anycast groups is 1 million.

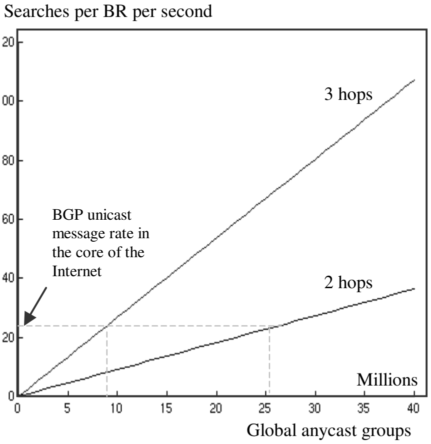


Figure 15: The Average number of search messages processed by a BR per second as a function of the total number of global anycast groups (in millions) and for the case where 0.5% of the domains have group members.

maintain a large unicast routing table, don't store any anycast routes. Routers in edge domains store routes to internal anycast groups and popular ones. The numbers of both group types are much smaller than the number of all global anycast groups in the Internet. Moreover, each edge domain can control the number of its internal and popular groups to stay within the limit of the locally available routing resources.

In addition, the fact that anycast addresses are distinguishable from unicast addresses means that anycast routes can be maintained in their own routing table separated from unicast routes. As a result, the existence of an anycast service does not slow down the unicast forwarding process. Moreover, the anycast routing table can use much simpler data structures and allow faster search and insertion than the unicast routing table because it does not need to account for the longest prefix match.

6.4 Processing Overhead at Border Routers

GIA's overhead is mainly located at border routers and is dominated by the processing of search messages. In this section, we show that the number of search messages generated by GIA is orders of magnitude less than the number of messages generated by routing anycast using the unicast routing protocols (the traditional approach.) Moreover, we show that the search overhead is small enough for the Internet to support millions of global anycast groups. Finally, we show that the interaction between unicast routing and anycast routing at a border router can be made minimal so that unicast routing is not affected by the existence of an anycast service. The simulations use the Internet inter-domain topology of November 1999, and the simulation environment described in Section 5.1. The number of messages generated by the traditional approach is computed by treating each anycast group as a unicast routing entry and using the information in [12].[10] Again, each data point in our graphs is the average of 100 runs.

Figure 13 shows that the number of search messages generated by GIA is orders of magnitude smaller than the number of messages generated by routing external anycast groups using the unicast routing protocol. There are three reasons why our design generates less control traffic than routing anycast using the traditional way. First, each domain searches only for its popular anycast groups. Second, a domain searches only its neighborhood. Third, once the route is learned it does not generate additional messages as long as the nearest member stays accessible. This is in contrast to routing anycast through BGP (without being GIA-enabled) in which case any change in the topology causes a cascade of routing messages.

Figure 14 shows the average number of search messages processed by a border router per second when the number of global anycast groups is 1 million. It reveals that for a maximum TTL of 2 or 3 domain hops, a BR processes only 1 to 2.7 messages per second. Figure 15 shows the average number of searches processed by a border router in a second as a function of the total number of global anycast groups.[11] In particular, the figure indicates that for a search rate equal to the current BGP message rate at the core of the Internet (23 messages/second [12]) the Internet can support 10 to 25 million global anycast groups. This indicates that the number of global anycast groups can grow quite large before it imposes a significant load on the routers. To further quantify this, Labovitz et al. report that in one scenario the routers were able to handle 70 routing messages per second [19]. This

---

[10] In our simulation, we use the measurements posted at [12], which shows the current number of BGP messages per unicast routing entry to be around 36 messages per day. Some researchers argue that the number of BGP messages for a routing entry increases exponentially with its connectivity [18]. Although this argument favors GIA over the traditional approach, our simulation ignores the effect of the high connectivity of a replicated anycast address on increasing the number of messages it generates when it is routed using the unicast inter-domain routing protocol (BGP).

[11] The graphs in Figure 15 are generated by scaling the results in Figure 14 for the case where 0.5% of the domains have members of each anycast group. ('0.5%' is a conservative choice because the fewer the domains that have members of an anycast group the larger the number of search messages.)

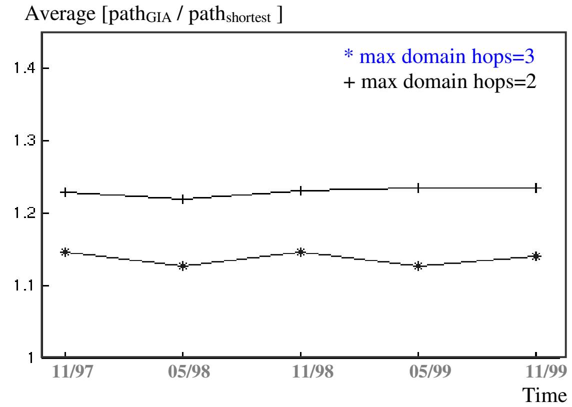


Figure 16: GIA's path efficiency is independent of the Internet growth

would allow the current Internet to support approximately 20 to 50 million global anycast groups. Note that the above estimate assumes that a search message has roughly the same processing cost as a unicast routing message. However, in reality, the cost of processing a search at an ISP is considerably less than that of processing a BGP update. Furthermore, the search cost incurred by an ISP is proportional to the benefits this ISP derives from the anycast service. More specifically, processing a search consists mainly of looking up the anycast groups solicited by the search in the anycast routing table. Since an ISP usually has few popular anycast groups, the cost of a search message at an ISP mainly depends on the number of internal groups in its network. However, an ISP would be paid to maintain members of internal groups in its domain. Therefore, the cost incurred by the ISP would be proportional to its benefits. Note too that the search rate can be controlled by the ISP. In particular, each ISP agrees with its client domains on a certain search rate. The ISP can easily monitor the number of searches received from each of its client domains and charge the client domain for the extra searches.

Finally, processing search and reply messages should not affect the BGP router performance and slow down its processing of unicast routing updates. To prevent this, BGP routers might assign higher priority to processing unicast updates. In fact, processing search and reply messages is logically independent from processing update messages (GIA requires only read access to the unicast routing tables), and can be performed by a separate CPU. In addition, searches are allowed to explore only a limited neighborhood around an edge domain. Thus, only a small number of search messages reach the backbones and most of them are processed by routers at the edges of the network where the traffic is not as intense.

6.5 GIA's Performance as a Function of the Internet's Growth

In this section, we show that the future growth of the Internet will not decrease the efficiency of GIA, nor will it hinder its scalability. Instead of taking each parameter of the Internet growth separately (e.g., number of domains or edge degree,) and study its effect on our protocol, we directly examine the combined effect of these parameters by plotting the performance as a function of time. We use 5 snapshots of the Internet inter-domain topology taken over a period of two years (see Appendix A1). We use the simulation environment described in Section 5.1, and we simulate the case

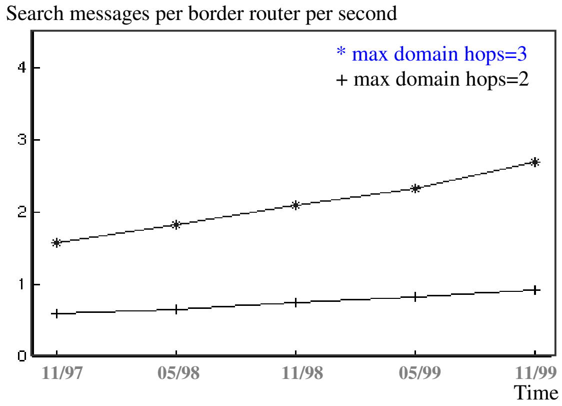


Figure 17: The number of search messages as a function of the Internet growth

where the fraction of domains that have anycast members is 0.5%.[12]

Figure 16 shows the average ratio of the path in GIA to the shortest path as a function of time. It indicates that GIA's path efficiency is not affected by the growth of the Internet. This is a significantly promising result. It means that we can maintain the efficiency at a constant and satisfactory level using only simple rules that do not change over time. (As the Internet grows we do not need to change the maximum TTL of a search from 3 to 4 domain hops nor do we need to change the scoping rules at transit domains.)

Next, to study the change in the search overhead as a function of the growth of the Internet we fix the value of the number of global anycast groups. Since we are interested in the trend rather than the exact numbers, the particular value we pick for this parameter is not important. Thus, we simulate the case where the number of global anycast groups is 1 million. Figure 17 indicates that the number of search messages processed by a border router increases linearly as the Internet grows. This linear increase is significantly slower than the increase in the CPU power at a border router. Moore's law suggests that CPU power increases exponentially with time. Although the routers might be slow in incorporating the advances in CPU technology, the large difference between an exponential and a linear increase indicates that the routers will be able to keep up with the increase in the search load resulting from the growth of the Internet.

7. IMPLEMENTATION AND OTHER ISSUES

In this section, we provide a brief description of our implementation, and discuss some issues that help constructing a complete understanding of the design.

7.1 A Prototype Implementation of GIA

To verify our design we extended the Multithreaded Routing Toolkit [21] to support a GIA-enabled border router. The current implementation works on a FreeBSD kernel and uses the experimental addresses in 10.0.0.0/8 as the anycast address space. It provides all of the functionality described in the above sections, and has been operationally verified in our laboratory's testbed.

---

[12] We fix the fraction of domains that have members to study the effect of the other parameters. (the exact value is not particularly important)

Our implementation has three building blocks: the Popularity-Monitor (PM), the Anycast-Routing agent (AR) and the Route-Maintainer agent (RM). It also involves a slight modification to the forwarding path in the kernel. For more details about the implementation please refer to [15].

7.2 Scoped Anycast Addresses

Although the previous sections focused on supporting global anycast groups, GIA does not exclude the use of scoped anycast groups. Scoped anycast addresses, as defined by IPv6 [13], can coexist simultaneously with global anycast addresses, as defined by GIA. They would be used for groups whose members are required to exist only in a scoped and relatively small region (e.g., the scoped anycast group representing the home agent for a mobile IP client).

7.3 Resilience to Failures

Resilience to loss of a learned route: A learned anycast route becomes unavailable when the domain, which has learned the route, loses connectivity to the nearest member or the nearest member crashes. Both of these cases have been discussed in Section 4.4.4.5. In this section we discuss two less common circumstances that affect the availability of a learned anycast route. First, a learned route becomes invalid when the border router at the end of the route crashes. In this case, the domain that has learned the route would keep tunneling the anycast packets to the failed router because it has no means of discovering the invalidity of its route. Although border router failures that last for a substantial period are not, and should not be, common events in the Internet, the design can be made resilient to such failures. To do so, we define the notion of a 'Border Address'. A Border Address is a unicast address shared by all the border routers in a domain. A Border Address need not be routed. Inside its domain, the Border Address is not used and need not be known. Outside its domain, the advertisement of the Border Address is aggregated into the advertisement of the domain's prefix (i.e., the Border Address gets free routing). When a border router replies to a search message, it includes its domain's Border Address in the reply. The domain that learns the route uses it by tunneling packets to the Border Address. As a result, the tunneled packets are delivered to the nearest border router in the domain of the nearest member of the popular group, regardless of whether this router is the one that sent the reply or not. Hence, tunneling anycast packets to the Border Address allows any BR in the domain of the nearest member to decapsulate the packets and deliver them to the local group member, which provides resilience to crashes of any particular BR.

Second, the destination domain of a learned route might be a sub-domain whose BGP updates are aggregated by its parent. If such a domain gets partitioned from its parent, the parent might not send a BGP update to withdraw the unicast address space of the child domain. Consequently, a domain that has a learned anycast route pointing to this partitioned child domain might not be notified about the partition, and would keep sending packets along the learned route. The problem can be solved by having a parent domain that receives an encapsulated anycast packet pointing towards a partitioned child, whose BGP advertisements are suppressed by the parent, decapsulate the packet, send it along the default route, and send an ICMP message to the encapsulating BR to inform it about the unavailability of the route.

Resilience to loss of the default route: A default route becomes unavailable if the domain loses connectivity to the home domain or the home member crashes. The architecture as described in the above sections provides mechanisms to both the anycast service provider and the client domain to considerably alleviate the impact of such failures. The anycast service provider can increase the resilience of its default route by replicating the service in the home domain or by providing the home member with some form of fault tolerance. In addition, an end domain that doesn't tolerate temporary loss of connectivity to a particular anycast group can explicitly configure its BRs to consider the group as a popular one. Since the mechanisms described in this section and in Section 4.4.4.5 render popular groups highly available, labeling a group as popular gives it a high resilience. Nonetheless, if the degree of resilience achieved through the above mechanisms is not sufficient and an ultimate resilience to losses of default routes is desired then the following scheme can be adopted. A router that receives a native (non-encapsulated) anycast packet and doesn't have a route to the home domain sends a special ICMP message towards the sender of the packet. In an IPv6 environment, the router addresses the ICMP message to the Subnet Router anycast address of the subnet of the sender of the anycast packet. (The Subnet Router anycast address as defined in [13] is an address that is shared by all routers attached to a link). In case the network is not IPv6 enabled, the router addresses the ICMP message to the sender of the anycast packet and includes the Router Alert option [16] in the packet's header. In either case, the local router on the sender's subnet receives this ICMP message and informs the border router in its domain of the unavailability of the default route. Depending on the domain's policy the BR might decide to search for the nearest group member or wait until the route becomes available.

8. DEPLOYMENT ISSUES

This section addresses incremental deployment in the Internet.

8.1 Changes to Routers

To deploy GIA in a transit domain we need to change the border routers to participate in route learning and to change the internal routers to shift the anycast indicator off when they have no route to the anycast group. However, changing the internal routers is not crucial. The same effect can be achieved by having the border routers inject the unicast inter-domain routing information internally after shifting the anycast indicator in. We propose this solution as an intermediate step until the domain upgrades the internal routers to understand the anycast address syntax.

On the other hand, deploying GIA in an edge domain requires integrating popularity monitoring, route learning, and route maintenance in the border routers. For most edge domains changing the internal routers is unnecessary because edge domains usually have only one exit point to the rest of the Internet and accordingly one border router. When the internal routers receive a packet addressed to an unpopular anycast group they treat it as a unicast packet for which they have no route; thus, they forward it to the border router. The border router, which is GIA-enabled, shifts the anycast indicator off and forwards the packet according to its unicast routing table. For the case of edge domains that have more than one border router, an intermediate stage similar to the one described for the transit domain case can be adopted.

If GIA is deployed in an IPv6 environment, the aforementioned changes can be incorporated to the routers while upgrading them to be IPv6 enabled.

8.2 Crossing Non-GIA-Enabled Regions

During the deployment phase, the Internet will contain both GIA-enabled and non-GIA-enabled regions. We would like a domain in a GIA-enabled region to forward packets addressed to an unpopular anycast group towards their home domain even if the home domain is separated from this domain by a region that is not GIA-enabled. One possible solution is to configure the border routers at the periphery of a GIA-enabled region to encapsulate anycast packets leaving the region in unicast packets addressed to the unicast address resulting from shifting the anycast indicator off. In addition, the border routers set the transport protocol field in the IP packet to a special protocol number that identifies these encapsulated anycast packets. The packets cross the non-GIA-enabled region safely heading toward the home domain. Once they enter another GIA-enabled region the border router recognizes them as encapsulated anycast packets. The BR decapsulates the packets, which then complete their path according to the scheme described in the above sections.

9. CONCLUSION

Although IP-anycast has long been defined and recognized as a useful service, its alleged unscalability has limited its acceptance by the community. This paper shows that it is possible to provide a scalable global IP-anycast. The results of simulating the proposed architecture on recent Internet topology indicate that the current Internet can easily support a few millions of global anycast groups. In addition, simulating the design on multiple snapshots of the Internet topology indicates that, despite its growth, the Internet will continue being able to support millions of global anycast groups. Finally, our implementation proves the practicality of the design.

The price to be paid to scale the service is a slight increase in the average path length. Particularly, the average path length in our architecture is 1.15 the path length resulting from routing anycast the traditional way using the unicast routing protocols. We believe that this slight decrease in the efficiency is not significant, and that the gained scalability far outweighs the overhead of the design.

10. ACKNOWLEDGMENTS

The authors would like to thank Mangesh Kasbekar and Saad Mneimneh for helping with the simulation, and Geoff Voelker for providing the organizational trace. We are also grateful to David Clark, Tim Shepard, and Steve Deering, who provided valuable insights. Thanks are also due to Chandrasekhar Boyapati, Charles Blake, Dorothy Curtis, Constantinos Dovrolis, and our Sigcomm reviewers for their constructive comments.

11. REFERENCES

[1] E. Basturk, R. Haas, R. Engel, D. Kandlur, V. Peris, and D. Saha, "Using Network Layer Anycast for Load Distribution in the Internet," *Proc. Global Internet'98* (1998).

[2] S. Bhattacharjee, M. H. Ammar, E. W. Zegura, N. Shah, and Z. Fei, "Application Layer Anycasting," *Proc. IEEE INFOCOM'97* (1997).

[3] M. Faloutsos, P. Faloutsos and C. Faloutsos, “On Power-Law Relationships of the Internet Topology,” *Proc. ACM SIGCOMM'99* (1999).

[4] Z. Fei, S. Bhattacharjee, M. H. Ammar, and E. W. Zegura, "A Novel Server Technique for Improving the Response Time of a Replicated Service," *Proc. IEEE INFOCOM'98* (1998).

[5] W. Fenner, "Internet Group Management Protocol, Version 2," RFC 2461 (1997).

[6] P. Francis, "Pip Near-term Architecture" (1994).

[7] P. Francis, S. Jamin, V. Paxon, L. Zhang, D. F. Gryniewicz, and Y. Jin, "An Architecture for a Global Host Distance Estimation Service," *Proc. IEEE INFOCOM '98* (1998).

[8] S. V. Fuller, T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation," RFC 1519 (1993).

[9] R. Govindan and A. Reddy, "An Analysis of Internet Inter-Domain Topology and Route Stability," Technical report USC-CS-96-642, Department of Computer Science, University of Southern California, *Proc. IEEE INFOCOM*'97 (1997).

[10] J. Gwertzman and M. Seltzer, “World Wide Web Cache Consistency,” *Proc. Usenix* (1996).

[11] R. Hinden, "Simple Internet Protocol Plus," RFC 1710 (1994).

[12] Internet Performance Measurements and Analysis (IPMA), http://www.merit.edu/ipma/trends/.

[13] R. Hinden and S. Deering, "IP version 6 Addressing Architecture," RFC 2373 (1998).

[14] D. Katabi, " The Use of IP-Anycast to Construct Efficient Multicast Trees," *Proc. IEEE Global Internet'99* (1999).

[15] D. Katabi and J. Wroclawski, “A Strategy and Protocol for Scalable IP Anycast,” MIT/LCS/TR-798 (2000).

[16] D. Katz, “IP Router Alert Option,” RFC 2113 (1997).

[17] D. Kim, D. Meyer, H. Kilmer, and D. Farinacci, “Anycast RP mechanism using PIM and MSDP,” Internet-Draft (2000).

[18] C. Labovitz, A. Ahuja, F. Jahanian, and A. Bose, "Experimental Measurement of Internet Routing Convergence," *NANOG'18* (1999).

[19] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet Routing Instability," *Proc. ACM SIGCOMM'97* (1997).

[20] K. Moore, J. Cox, and S. Green, "Sonar - a Network proximity Service," Internet-Draft (1996).

[21] The Multi-threaded Routing Toolkit (MRT), http://www.mrtd.net.

[22] A. Myers, P. Dinda, and H. Zhang, “Performance Characteristics of Mirror Servers on the Internet,” *Proc. IEEE INFOCOM'99* (1999).

[23] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461 (1998).

[24] The National Laboratory for Applied Network Research (NLANR), http://www.moat.nlanr.net/AS/.

[25] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," RFC 1546 (1993).

[26] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771 (1995).

[27] S. Seshan M. Stemm, and R. Katz, "SPAND: Shared Passive Network Performance Discovery," *Proc. USITS '97* (1997).

[28] A. Wolman, G. Voelker, N. Sharme, N. Cardwell, M. Brown, T. Landray, D. Pinnel, A. Karlin, and H. Levy, “Organization-Based Analysis of Web-Object Sharing and Caching,” *Proc. USITS* (1999).

Appendix A1: Simulation Graphs

http://moat.nlanr.net/Routing/rawdata/Asconnlist.19971110.879158401
http://moat.nlanr.net/Routing/rawdata/Asconnlist.19980510.894793200
http://moat.nlanr.net/Routing/rawdata/Asconnlist.19981110.910694401
http://moat.nlanr.net/Routing/rawdata/Asconnlist.19990510.926329200
http://moat.nlanr.net/Routing/rawdata/Asconnlist.19991108.942057661